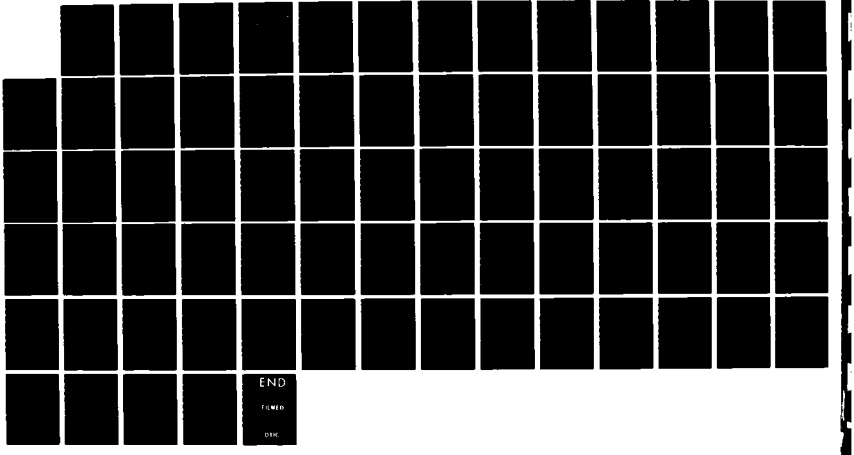
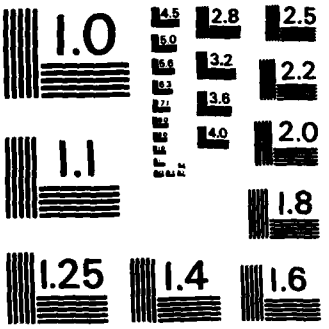
AD-A160 364 A BASIC GUIDE TO UNIX(U) UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST L MOSES OCT 85 ISI/TM-85-157 MDA903-81-C-0335 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

ISI Technical Manual
ISI/TM-85-157
October 1985

Lisa Moses

University of Southern California

A Basic Guide to UNIX

INFORMATION SCIENCES INSTITUTE

213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/TM-85-157	2. GOVT ACCESSION NO. AD-A160	3. RECIPIENT'S CATALOG NUMBER 364
4. TITLE *(and Subtitle)* A Basic Guide to UNIX		5. TYPE OF REPORT & PERIOD COVERED Technical Manual 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lisa Moses		8. CONTRACT OR GRANT NUMBER(s) MDA903 81 C 0335
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90292-6695		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE October 1985 13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

This document is approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

.....

18. SUPPLEMENTARY NOTES

.....

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

CCA Emacs, csh, Diablo printer, FTP, HM, MH, MM, Scribe, Tcsh, Telnet, UNIX

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This basic instructional manual introduces the use of several programs that are used on the DARPA sponsored UNIX 4.2 bsd computer systems at USC/Information Sciences Institute. The programs covered include the CCA Emacs text editor, the MH/HM and MM message programs, the Scribe formatting program, and some csh and Tcsh commands. This manual is intended for people who may not have experience with UNIX or with programming.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ISI Technical Manual
ISI/TM-85-157
October 1985

University of Southern California

Lisa Moses

A Basic Guide to UNIX

INFORMATION SCIENCES INSTITUTE
213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

This research is supported by the Defense Advanced Research Projects Agency under Contract No. MDA903 81 C 0335. Views and conclusions contained in this report are the author's and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any person or agency connected with them.

Acknowledgments

Thanks to Ray Bates and Tom Wisniewski for patiently explaining some of the intricacies of Unix, and to Lewis Johnson for reviewing this manual and giving me really helpful feedback. Also, thanks to Victor Brown and Sheila Coyazo for their editing skills.

Table of Contents

1. INTRODUCTION

This is an introductory document for UNIX[1] (version 4.2) on the ISI VAX machines. It is intended for people who would like to use UNIX for electronic mail, text editing, and document preparation, but who may not have experience with UNIX or with programming. It does not include an in-depth explanation of the programs covered; instead, it is a guide to the basic commands.

With the programs presented here, there is often more than one way to achieve the same results. I introduce the simplest set of commands to carry out a task, the assumption being that the user may proceed to more complex documentation after learning these basic commands.

In this document, the underlined words are what ***you*** type to execute commands. Underlined words that are also italicized are not meant to be typed literally, but represent individual names. For example, when you see

login: *directoryname*

you should type your own directory name rather than the word "directoryname." Another word that you will often see (underlined and italicized) is "filename." You must always substitute the actual name of the file that you are working with for the word "filename."

Generally, the System's response to your typed command is not shown, but only each sequence that you must type.

The instructions for most programs are divided into "lessons." Each lesson should be absorbed and understood before progressing to the next one. If this is your first experience with these programs, you should review all of these lessons periodically until you understand them completely.

---

[1] Copyright © Bell Telephone Laboratories, Incorporated.

In this document, you will find repeated references to the **Return**, the **Control Key**, and the **Escape Key**. Since they are of such importance to the user, they have been assigned specific symbols.

- The **Carriage Return** is represented by ↵. Most commands are not executed until you type a carriage return. Often a command given by the user must be confirmed before it is carried out by the System. The carriage return is used to confirm a command. If you type a command and the cursor sits blinking after the last character typed, you probably need to type a carriage return.

- The **Control Key** is represented by ↑. It is used with another key to enter a command (similar to the Shift Key). Hold the Control Key down then press the other key and release it immediately.

- The **Escape Key** is represented by ⇐ and the Escape Key is used in Emacs, HM, MM, and the Tcsh. It functions differently depending on the program in which it is used.

2. DEFINITIONS

CURSOR The Cursor is the small blinking line (about the size of a hyphen) that indicates your position on the terminal screen.

DEFAULT Whenever the System does something that is **assumed**, rather than something you've specified, it is a default. For example, if you give the command to see the contents of a file by typing cat *filename*, the System will assume that the file is in your directory. That is the default. If you want to see a file in another directory, you must give the System more information.

DIRECTORY

A directory is a collection of files. Your directory is your working space on the System. You can think of it as your office with your files in it. When you login (identify yourself to the System by giving your directoryname and password), you are given access to your directory and files. You may also create subdirectories within your directory, in order to group related files together. The term "home directory" refers to the directory you are automatically connected to when you login.

FILE A file is a specific collection of data such as a letter, a list, or a report. Files have names to enable the computer to recognize them. A file can be long (an entire document) or short (one line).

FILENAMES

Files must be given names when they are created. A filename may consist of one or more words and may have up to 256 characters. Filenames are often composed of two parts: **name.type** . These parts are separated by a period. Usually the type consists of only one word, but the first part may consist of several words separated by hyphens. **No spaces are permitted in filenames.** When you name a file, it is important to choose a name that is unique and will also remind you of what is in the file. The filename type may indicate the kind of file it is. This is particularly true with files that contain Scribe (a formatting program) commands. UNIX is **case sensitive**, meaning that if your filename contains any uppercase letters, you must type the filename exactly as you have named it for UNIX to recognize it.

There are no generation or version numbers with UNIX files. Whenever a file is updated. or given the same name as another file, it ***replaces*** that file and the file that originally had the name is removed.

PASSWORD

Your password is the key to your directory. When you login, you type your directoryname and then your password. When you type your password. it will not appear on the terminal screen. This way, you may keep your password a secret. Your password should be approximately eight characters long.

PROGRAMS

Programs issue instructions to the computer. Some programs process files. For example, a **text editing program** can follow commands to create a text file and edit it, while a **text formatting program** can follow commands to format text by centering a heading, enumerating a list. or italicizing a name. Programs are stored as files on the System.

SWITCHES

Switches are options that may be given with commands. They are preceded by a hyphen. For example, to see a list of the names of the files in your directory type ls). To see the same list with additional information (size of file, date and time it was last changed, and owner) type ls -l) . The -l is a switch.

UNIX SHELL (csh)

UNIX is the main program you use to communicate with the computer. It interprets all of the commands and programs that you use, controls and allocates resources, and schedules and prioritizes jobs. UNIX is referred to as the **Operating System** or the **System**. When you login, you are using a program called the UNIX **Shell** or **csh**. From the Shell, you can enter different programs, and when you leave a program you return to the Shell. You logout from the Shell. The actual Shell name is "csh" but will also be referred to as the Shell in this manual. When you are in the Shell, you should always see a "%" sign on the left side of the screen, preceded by a number. This is the Shell prompt, and the number indicates the number of commands you have given since you logged in.

TREE STRUCTURE/PATHNAMES

Directories on the UNIX System are arranged in what is known as a "tree structure." All directories branch out from a common directory called the **root** directory. The root directory is represented by / (a forward slash). This tree formation is the reason you must specify a **pathname** before the System will allow you to work in or copy files from another directory.

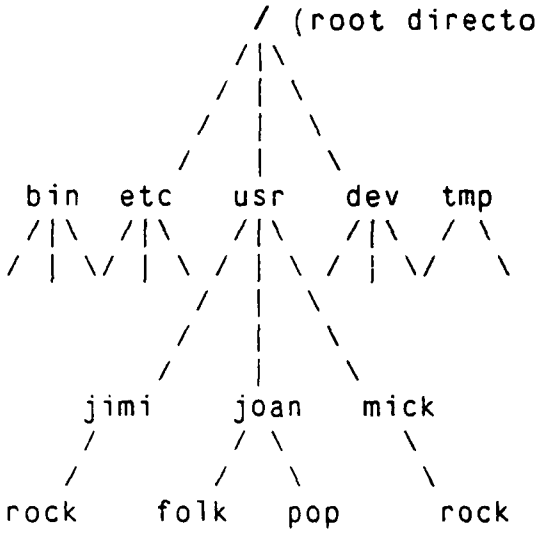


Figure 2-1: Example of a Tree Structure

In the illustration above, user **jimi** has a file named **rock** and user **mick** also has a file named **rock**. However, in this example, these two files are unrelated to each other. If jimi wanted to see his own file, he would type cat rock↵, but if he wanted to see mick's file by that name, he would type cat /usr/mick/rock↵ . This sequence is called the file's pathname. Looking further up the structure, you can see that both mick and jimi's directories are under the directory /usr.

3. UNIX – BASIC SHELL COMMANDS

Key:
↑ = Control Key
) = Carriage Return

3.1. UNIX – Lesson One – Getting Started

SHELL PROMPT The % is the prompt for the UNIX Shell. In order to give any commands to the UNIX Shell, you must have the % prompt. The % is preceded by a number. When you first login, you will see 1%. This indicates the System is ready to accept your first command. After typing a command, you will see 2%. The number indicates the number of commands you have given since you logged in.

LOGGING IN To login to the System, type at the "Login:" prompt *directoryname*)*password*) (type in lowercase). Your password will not appear on the screen when you type it. You must login to the System before you can do any work. There will be a pause after you login before you get the UNIX prompt (1%). Do not type ahead during this pause or between your login name and password.

LOGGING OUT To logout, type logout) . When you are finished working for the day, you should leave the System by using the logout command.

ABORTING A COMMAND

Type ↑C to abort a partially typed or partially completed command.

LISTING FILENAMES

To see a list of the files in your directory, type ls) . Several options you may use to list filenames with this are:

ls –a Type ls –a to see a list of all filenames. Filenames that start with a . (period) won't show unless this switch is used.

ls –l Type ls –l to see a list with information on each file. (owner, size, date, and time it was last changed).

ls –t Type ls –t to see a list of files in the order of their creation, with the newest ones first.

CHANGING YOUR PASSWORD

To change your password type
passwd↵ *oldpassword*↵ *newpassword*↵ *newpassword*↵ .

FREEZING THE SCREEN & RESTARTING

Type ↑S to freeze the screen to read a file or message that is longer than one screen. To restart, type the space bar.

DELETING A CHARACTER

Type ← to delete a character to the left of the cursor on the IBM PC. On other terminals, use the Del Key. You may need to use the Shift Key with the Del Key.

DELETING A WORD

Type ↑W to delete a word to the left of the cursor.

DELETING A LINE

Type ↑U to delete a line from the cursor to the left margin.

HISTORY

To see the last 20 commands you have typed to the Shell. type history↵ . To repeat a history (previous) command, type !#↵ (# is the number of the command next to the Shell prompt).

LOAD AVERAGE

To see the load average, type w↵ . The load average is a measure of how busy the System is. If it is under four you can expect a quick response to your commands since the System is not very busy. If it is over four, you may notice a slower response. The w command shows three load averages. The first one is for the last minute, the second is for the last five minutes, and the third is for the last fifteen minutes. This command also shows who else is logged in.

TYPING COMMANDS IN LOWERCASE

UNIX does notice when you type in upper or lowercase. You must type most commands in lowercase. However, if a command is shown as uppercase in the documentation, then you must type it in uppercase. When you are naming a file be aware that if you type any part of the name in uppercase, you must ALWAYS type the filename exactly that way. For this reason, it is probably better to type filenames in lowercase. **Be particularly careful to type in lowercase when you login.**

FINGER

Use finger to see information about another user (full name, phone, office number, etc.). Type finger *directoryname*↵ .

There are two files in your directory that finger will consult for information about you, for the purpose of showing others when they use finger on you. You may edit these files in Emacs:

.project	This should have a one-line description of the project you are working on.
.plan	This may have your vacation plans, work schedule, or where you can be reached. It may be as long as you want.

HELP

To read a description of a command, type man *command*↵. If you do not know the exact command, you may type apropos *word*↵ and a list of possible commands relating to the *word* will be printed. For example, to find out all the commands related to the word *directory*, type apropos directory↵ . Two of the commands that are displayed are

ls	list contents of directory
mkdir	make a directory

To see a full help description of each command, type man ls↵ or man mkdir↵ .

TUTORIAL

There is an on-line tutorial provided with Berkeley UNIX. To work with it, type learn↵ . There are several topics covered, but the most important ones to start with are "files" and "morefiles."

3.2. UNIX - Lesson Two - Handling Files

LOOKING AT FILE CONTENTS

Use cat, more, head, and tail to look at the contents of a text file.

cat *filename*↵ see the entire file without stopping

more *filename*↵ stop at each full screen (24 lines)

head *filename*↵ see the first 10 lines of a file

tail *filename*↵ see the last 10 lines of a file

COPYING A FILE FROM ANOTHER DIRECTORY

To copy a file from another user's directory type
cp *~directoryname/filename filename*↵
For example, to copy a file called **rock** from user **jimi** type
cp ~jimi/rock rock↵. (The ~ indicates the home directory.)

CREATING A SECOND COPY

To create a second copy of a file, type cp *oldfilename newfilename*↵.

RENAMING A FILE

To rename a file, type mv *oldfilename newfilename*↵.

REMOVING A FILE

To permanently remove a file from your directory, type rm *filename*↵ (this deletes and expunges the file). I recommend the alias **del** (see Appendix I. Hobgoblin) for deleting files because it puts the deleted file in a temporary directory from which it may be undeleted with the **und** alias or expunged with the **exp** alias.

SEARCHING IN FILES

You may search through a file or set of files for a string of text by typing grep *'string' filename filename filename*↵.
For example, to search for the string **as time goes by** in the files **song-standards**, **songtitles**, and **song-ballads**, type
grep 'as time goes by' song-standards songtitles song-ballads↵.

CORRECTING SPELLING ERRORS WITH ISPELL

After creating a file in Emacs, check it for spelling errors with ispell. Type ispell *filename*↵ . Ispell checks each word in your file against a dictionary and questions you about words that it doesn't find in that dictionary. It shows you each line with the word in question highlighted, so you can see the word in context. You may also build your own dictionary as you correct files. Ispell will name your dictionary **ispell.words**. Type a ? to see the commands to use in ispell.

PRINTING FILES

LINEPRINTER	print *filename*↵
PENGUIN	
print double-sided copy	xp *filename*↵
print single-sided copy	xp1 *filename*↵
see print queue	dpq↵
cancel print request	dprm *number of queue entry*↵
IMAGEN PRINTER	
type this line at the Shell (or enter it in your .cshrc file) in order to use the Imagen	setenv PRINTER isi-imagen↵
print single-sided copy	lpr filename↵
see print queue	lpq↵
2700 PRINTER	
print single-sided copy	hprint *filename*↵
print landscape mode	hprint –l *filename*↵
see print queue	lpq -Plp↵

ARCHIVING & RESTORING FILES

You may archive a file (store it on tape, off the computer) and you may also retrieve an archived file.

arch To archive a file, type arch –r *filename*↵ . The –r switch causes the file to be deleted in your directory when it is archived.

thaw To retrieve an archived file, type thaw↵ . You will be shown a list of your archived files. Type the number shown at the left of the filename to have it restored to your directory. Type h↵ to see all the available commands.

FILE PROTECTION

Not everyone can access everyone else's files. File access is determined by the file's protection status. You may control others' access to your files by granting or denying privileges. The privileges are: being able to read a file, write a file, and execute a file (if a file is a program, execute means you can run the program).

SEEING THE FILE PROTECTION

To see the file protection for all of your files, type ls -l↵ . The typeout will look like this:

```
-rw-r—r— 1 moses    358 Nov 18 14:31 header.mss
-rw-rw-rw- 1 moses  90909 Dec 13 10:47 intro.mss
```

The 10 characters on the left indicate file protection. The first character is reserved for the System. The next 3 refer to you (the user), the next 3 refer to your group, and the last 3 refer to all other users.

A privilege is represented by one of the following characters: the r means read access, the w means write access, the x means executable. The - means the privilege is denied. The protection shown for the first file listed above, **header.mss**, indicates that the user has read and write access, the user's group has read access, and all others have read access. The protection shown for the second file indicates that everyone has read and write access.

CHANGING FILE PROTECTION

Use the command **chmod**. Indicate whose access you are changing by typing one of the following letters:

u = yourself, g = your group, o = all others

Then indicate whether you are removing or adding a privilege by typing a + (plus) or a - (minus). For example, to set the file protection of the second file shown in the example above so that only the user and the user's group has write access and others do not, type

chmod o-w intro.mss↵

The - removes the privilege and the + grants a privilege. So typing o-w removes others' (o) write access (w). To grant write access to others, type

chmod o+w intro.mss↵

3.3. UNIX - Lesson Three - Subdirectories

Subdirectories are used frequently in UNIX for keeping related files together. You may easily create subdirectories and move back and forth from your home directory to any of your subdirectories.

~ A tilde refers to your home directory.

.. Two periods refer to the directory one level up in the tree.

. A single period refers to the directory you are currently in.

CREATING A SUBDIRECTORY

To create a subdirectory, type mkdir *newsubdirectoryname*↵.

CONNECTING TO A SUBDIRECTORY

To connect to a subdirectory, type cd *subdirectoryname*↵. When you connect to a subdirectory, any commands you give will work on the files in that subdirectory.

CONNECTING TO YOUR HOME DIRECTORY

To return to your home directory after connecting to a subdirectory, type cd↵.

SEEING WHERE YOU ARE

To find out what directory you are connected to, type pwd↵ (pwd stands for **print working directory**). This also shows your path in the tree.

MOVING A FILE TO A SUBDIRECTORY

To move a file from your home directory to a subdirectory (when you are in the home directory), type mv *filename subdirectory*↵. If you are in the subdirectory, type, mv ~/*filename* .↵.

DELETING A SUBDIRECTORY

Type rm -f *subdirectory* to delete a subdirectory. This will also delete any subdirectories below it. Use this command with caution because you cannot reverse it.

SEARCHING FOR A FILE

To search through your directories for a file, type (from your top level directory) find . -name *filename* -print↵. To see the names of all your .mss files, type find . -name *.mss -print↵.

3.4. UNIX - Lesson Four - Jobs

UNIX allows you to run more than one program at once. When you run a program it is considered by UNIX to be a **job** or **process** and it is given a number. In this chapter, the *#* means the *job number.*

STOPPED JOBS

When a program (like Scribe or HM) is ended with ↑Z rather than with an exit command, it is called a **stopped job**. This means that the program is in a suspended state and may be continued from exactly where it was when it was stopped. When the program is continued, it may be run in the foreground (as it would normally run), or you may have it run in the background so you can do other work while the program is running.

SEEING STOPPED JOBS

To see what jobs are stopped, type jobs↵.

CONTINUING A STOPPED JOB

To continue a stopped job, type fg↵. If you have more than one stopped job, type fg %#↵ (# is the number of the stopped job).

KILLING STOPPED JOBS

Type kill %#↵. Stopped jobs should be killed before logging out.

RUNNING A BACKGROUND JOB

You may run a job in the background while continuing with other work. Start the job (for example, start a Scribe job by typing scribe *filename*↵), type ↑Z to stop it, then immediately type bg↵. The output of the job you are running in the background will still appear on your screen.

You may also run a background job and redirect the output to a file. To do this, you must set it up when you initially run the job. For example, to run a Scribe job on a file called test.mss, type

scribe test.mss >& log &↵

This will create a file called log containing the output from Scribe that would normally print on your screen.

NOTE: A job running in the background will continue to run even if you logout.

3.5. UNIX - Lesson Five - Fancy Features

ALIAS An alias is a word that may be used in place of a UNIX command to accomplish that command. You may set your own aliases to be words that are more familiar to you than the actual UNIX commands are. For example, the UNIX command to look at the contents of a file is cat *filename* but you may be more comfortable using type *filename*. In that case you can set "type" to be an alias for "cat" and then either word will accomplish the same command. Create this alias by adding the following line to your .cshrc file (located in your home directory):

alias type cat

An alias may also represent a group of commands. To see what commands have alias names set for you, type alias↵ .

REDIRECTING OUTPUT FROM A PROGRAM TO A FILE

Use the symbol > to redirect output from a program into a file, rather than having it print on the screen. For example, type ls >filenames↵ to create a file called **filenames** that contains the output of the ls program (a list of the files in your directory). With this method, a file is always created.

PIPES (USING PROGRAMS TOGETHER)

The pipe symbol | connects the output of one program to the input of another program allowing you to combine the processes of each program. For example, type ls | print↵ to send the output of the ls program (a list of the files in your directory) to the lineprinter. With this method, a file is not created.

CORRECTING COMMAND STRINGS AT THE SHELL

It is possible in UNIX (particularly with the use of the ! and the >) to give complex command strings to the Shell. If you make a typo, you may use the history command to point to the string and then correct it. For example, to have the output of **scan** go into a file called **tmp.msg** you might accidently type

6% **scanm >tmp.msg**

To correct the word scan, you could type

7% !6:s^scanm^scan^↵

The !6 points to the number of the command, the : separates the number, the **s** means substitute, and the ^ separates the existing text (or typo) from the desired text. Once you complete the substitution, the command will be done immediately.

WILDCARD CHARACTER

A wildcard character is one that allows you to refer to more than one file at a time. The * (asterisk) matches any sequence of characters in a segment of the filename. For example, if you have files called **letter.emily** and **letter.gavin**, print them both at once by typing print letter.* . The * is used to indicate any part of a filename and can be used several times in a name. The * by itself indicates all filenames in a directory.

4. TCSH - USING AN ALTERNATE SHELL

The tcsh is a different version of the standard Shell (csh). It has all of the features of the standard Shell, and also some additional features that are described below. To use the tcsh, type (at the Shell) chsh *directoryname* /bin/tcsh↵ then login again. Now you will always get the tcsh Shell when you login. Type man tcsh to see a complete description of the tcsh Shell.

COMPLETING FILENAMES

Use the Escape Key to complete filenames at the Shell. Type enough of a filename for the System to distinguish it from your other files, then type the Escape Key to complete it. This only works at the Shell.

SEEING FILENAMES

Type ? to see a list of filenames while you are typing a command. For example, if several files start with the word **letter**, and you want to see the contents of one of them but don't remember the complete name, you may type ls letter? and a list of all filenames in the connected directory that start with **letter** will be printed on the screen.

EDITING SHELL COMMANDS

You may use these commands to edit commands at the Shell :

↑B	Move back one character
↑F	Move forward one character
↑A	Move to beginning of line
↑E	Move to end of line
↑D	Delete character cursor is directly under
← (or Del)	Delete character to the left of the cursor
↑W	Delete previous word
↑P	Move to previous command line. You may repeat a command that was given previously, by moving backwards through your commands with ↑P. Type ↵ when you reach the command you want to repeat. You may also edit the command before you type the ↵.
↑N	Move to next command line. If you typed ↑P to move to a previous command, you may use ↑N to move forward through your commands. ↑N does not cause a command to be repeated unless ↵ is typed.

5. MH - BASIC COMMANDS

MH is a group of commands that are given from the UNIX Shell. They are designed to create and manipulate messages. Each command is actually a separate program and some of these programs interact well with other Shell programs. The commands included here will allow you to create, send, read, and manage your mail on UNIX.

Figure 5-1 below shows a typical message. Each part of the message is called a **field** (To field, Subject field, Date field, etc.). The fields shown in this particular message (ending with and including the To field) are what make up the message **header**. Sometimes, you may wish to see only the header information in a message. When you give the command for this (scan), the header information will be shown on one line.

```
(Message inbox:22)
Date: 11 Sep 1984 08:21:06 PDT
Subject: Today's Meeting
From: BGEORGE@ISI-VAXA
To: TTURNER@ISI-HOBGOBLIN
cc: AJARREAU@ISI-HOBGOBLIN. BGEORGE@ISI-VAXA

We will be meeting today at 2:30 to discuss current
production expenses and to plan the next tour.  Plan to
stay through dinner.
-------
```

Figure 5-1: Sample message

A complete MH manual, "The MH Message Handling System," R-2367-AF, is available from Rand Corp., Santa Monica, CA.

NOTE: The MH and HM mail programs work on a file called **inbox**. When you read your mail with MH or HM it is automatically filed in **inbox** and can be read by either MH or HM. The MM mail program works on a file called **mail.txt**. When you read your mail with MM it is automatically filed in **mail.txt** and can be read ONLY by MM.

5.1. MH – Lesson One – Reading Mail

Key:
↑ = Control Key
= Message Number
↵ = Carriage Return

WHERE YOUR MESSAGES ARE

Message files are called "folders." The folder called +inbox contains your current messages.

NEW MAIL (inc)

When you receive a new message, the notice "you have new mail." will appear at the Shell. (This notice appears when you exit from a program or type ↵ at the Shell.) You must type inc↵ in order for the new message to be incorporated into your +inbox so that you may read it. You may also use this command to see if you have new mail.

SEEING MESSAGE HEADERS (scan)

To see your message headers (message number, date, sender, subject) in the order that the messages arrived, type scan↵. To see the last 10 (most recent) messages type scan ¦ tail ↵. This is an example of a Shell command (tail) interacting with an MH command by using a pipe (¦).

READING A MESSAGE (show)

To read a message, type show followed by the message number (for example, show 5↵ to see message 5). You may read your next message by typing next↵ and read your previous message by typing prev↵.

SEARCHING FOR MESSAGES (pick)

The pick command allows you to scan or show only certain messages. For example, type pick -from dbowie -scan to see the headers of messages from dbowie.

To find messages in another folder, type
pick -src *+foldername* -from *name* -scan
The folder you search in will become your current folder.

FINDING MORE INFORMATION ON COMMANDS

A complete list of MH commands can be obtained by typing man mh↵. To read in-depth information about a command, type man followed by the command. For example, type man scan↵ for information on the scan command.

ABORTING A COMMAND

Use ↑C to stop the printout of a message or message headers.

NOTE: For commands that require a message number, you may also give several numbers separated by a space. For example, type show 1 3 7 to show messages 1, 3, and 7. For consecutive messages, type the first message number and the number of messages, separated by a colon. For example, scan 5:7 would print 7 messages starting with message number 5.

5.2. MH - Lesson Two - Sending a Message

The comp command is used to create a draft message. When you type it, you will immediately enter CCA Emacs (see Chapter 8) to create the entire message. To use the comp command, type the UNDERLINED words as follows:

comp) (Once you enter Emacs, type ↑E to position the cursor)

To: *TTurner@isi-hobgoblin* ↑N	To send a message to more than one person, separate each directoryname with a comma. You may capitalize directorynames. If there are more names in the To field than will fit on one line, then start each new line with one blank space.
Cc: *BGeorge* ↑N	This is for carbon copies. Type the directorynames of those you want to receive a carbon copy of the message (including yourself).

Subject: *Today's Meeting* ↑N↑N

Text:
We will be meeting today at 2:30 to discuss current production expenses and to plan the next tour. Plan to stay through dinner.

↑X↑S	To save your message.
↑X↑Z	To end your message.

When you see "what now" you have the following options:

s)	send the message
e)	return to Emacs with the message
e ispell)	enter ispell with the message
q)	quit from (abort) the message
l)	list (redisplay) the message

WHEN THERE IS AN EXISTING DRAFT MESSAGE

If you start a message and then abort it, the next time you type **comp** you may see:
"/usr3/name/Mail/draft" exists; delete?

Your options are:

no↵	leave draft intact - don't send new message
yes↵	erase draft and create new message
use↵	edit existing draft
list↵	see existing draft

ADDING A FROM FIELD

If you are sending a message for someone else, you may add a From field to your header showing their name. To do this, insert a blank line after the Cc field, type From:, then their directoryname followed by their net address (for example, From: Ryland@VAXA).

INSERTING A FILE IN A MESSAGE

You may insert a file that was created with another program (like Emacs) into a message that you are creating, by placing the cursor at the point you want the file to be inserted and typing .

^x insert file *filename*↵

REPLYING TO A MESSAGE (repl)

To reply to a message you have received, type repl #↵. You will immediately enter Emacs and you may complete and send the message. Some header information will already be completed for you (although you may edit it). **NOTE:** The default with this command is to include everyone who received the original message in the cc: field. **Be sure to check the cc: field and delete names of those you do not want to receive your reply.**

FORWARDING A MESSAGE (forw)

To send a copy of a message you have received to someone else, type forw #↵. You will immediately enter Emacs and you may add your own message at the top of the message you are forwarding, and then send it.

NOTE: With both the **reply** and **forward** commands, you don't need to specify the message number if you have just read the message you are replying to or forwarding.

5.3. MH – Lesson Three – Managing Your Mail

FILING MESSAGES

This command is useful for collecting related messages into one folder. To move messages from your +inbox into another message folder, type file # +*foldername*↵.

SEEING FOLDER NAMES

Type folders↵ to see a list of the names of all of your mail folders.

CHANGING FOLDERS

To read a folder other than +inbox, type folder +*foldername*↵. Now any MH commands you use will work on that folder.

REMOVING A MESSAGE

To remove a message, type rmm #↵. Messages removed with this command may not be retrieved. They are really gone.

REMOVING A FOLDER

To remove a message folder and all the messages in it, type rm –r Mail/*foldername*↵. Use this command with caution because you cannot reverse it.

RENUMBERING A FOLDER

After you have removed messages from a folder you may type folder –pack↵ to cause the folder to be renumbered consecutively.

PRINTING A HARD COPY

To print a message on the lineprinter, type print Mail/inbox/#↵ (# is the message number). You may print more than one message at a time. For example, to print messages 9, 15, and 28, type print Mail/inbox/{9,15,28}↵. To print a message on the Penguin, use **xp** instead of **print**.

Another method of printing a message is a two-step process.

1. Type show # # >tmp.msg↵. This will create a file called tmp.msg with your message(s) in it.

2. Type print tmp.msg↵ to print tmp.msg on the lineprinter.

6. HM - BASIC COMMANDS

HM is a Rand message program that allows you to work in a split-screen environment with the message headers in your current folder in the top third of the screen. When you read a message, it is displayed in the bottom two-thirds of the screen. Most commands are given with a single letter and work on the message that the cursor is positioned next to. When mail is deleted with HM, it is moved into a folder called **+deleted**, and may be moved back into +inbox (or any other folder), or permanently expunged. HM and MH work on the same mail folders, so you may use them interchangeably to work with your mail. MH has more commands available, but HM is faster to use with large amounts of mail. Enter HM by typing hm↵ at the Shell.

LIST OF HM COMMANDS

h	see complete list of commands
i	incorporate new mail into +inbox
t	type message
s	enter Emacs to compose and send message (see page 20)
r	reply to message
f	forward message
m	move message to another folder. If this is a new folder, HM will ask "Create new folder "*foldername*"? (y or n): ". Type y↵ to create a new folder and move the message next to the cursor.
c	change folders. Use this to see a message in another folder.
L	type list of folder names
d	delete message (moves message to +deleted folder)
x	expunge deleted messages
p	pack folder (use after moving or deleting messages)
↑L	clear screen and reprint it
↑C	abort HM command
q	quit from HM

LIST OF HM/EMACS CURSOR COMMANDS

↑N	Move to next line
↑P	Move to previous line
⇗<	Move to beginning of folder
⇗>	Move to end of folder
↑V	Display next screen of headers
⇗V	Display previous screen of headers

7. MM - BASIC COMMANDS

MM[2] is a program used to create and manipulate messages. The commands included here will allow you to create, send, read, and file your mail.

MM commands may be abbreviated. In the following lessons, I have underlined the part of each command that must be typed.

The figure below shows a typical message. Each part of the message is called a **field** (To field, Subject field, Date field, etc.). The fields shown in this particular message (ending with and including the To field) are what make up the message **header**. Sometimes, you may wish to use one of the header commands to see only the header information in a message.

```
Msg 27 (520 chars):
15-Aug-85 09:24:46-PDT,325:000000000000
Received: by isi-hobgoblin.ARPA (4.12/4.7)
From: Mercury (Freddy Mercury)
Date: 15 Aug 1985 0924-PDT (Thursday)
To: Sting
Cc: Mercury
Subject: Live Aid

Dear Sting,

I saw your act at the Live Aid concert...you were great!
Branford Marsalis was especially wonderful on the saxaphone
(as always).  Let's get together soon.

Freddy
```

Figure 7-1: Sample message

More complete MM documentation is available on ISIB in Comp:<Documentation>MM.Changes, MM.Doc, and MM.Info. This documentation is, however, intended for MM on TOPS-20 and some of the features covered may not yet be available on UNIX.

---

[2]MM was written by Michael McMahon of Stanford University. with contributions from several other individuals. MM was originally written for TOPS-20 and has been developed for UNIX by David L. Kashtan of SRI.

7.1. MM – Lesson One – Reading Mail

Key:
↑ = Control Key
⇐ = Escape Key
↵ = Carriage Return
= Message Number

MM Enter MM from the Shell by typing MM↵ .

MM> MM> is the MM prompt. Whenever you see MM> on the left side of the screen, you are in MM.

WHERE YOUR MESSAGES ARE

The file mail.txt contains your current messages. It is created when you receive your first message. When you enter MM, this file is automatically loaded into MM for you to read or modify.

SEEING MESSAGE HEADERS

To see your message headers (message number, date, sender, and subject) in the order that the messages arrived, type headers all↵ . To see your headers in reverse order (most recent messages first), type headers inverse↵ . To see other commands that can be used with headers, type headers ?↵ , and MM will print a list.

READING A MESSAGE

To read a message, type type followed by the message number (for example, type 5↵ to see message 5).

LEAVING MM To leave MM and return to the Shell, type quit↵ .

ABORTING To abort the typeout of your headers or of a message, type ↑Q↵. To abort a command, type ↑N.

CUSTOMIZING MM

The file called mm.init allows you to customize some aspects of MM. Create this file by entering MM and typing create⇐↵ . Exit MM, and use Emacs to edit mm.init.

DELETING A CHARACTER

Type ← (or Del Key) to delete the character to the left of the cursor.

DELETING A WORD

Type ↑W to delete the word to the left of the cursor.

? Type ? to see the commands that are available at any point in MM, and a list will be printed on your screen. Once the list is printed, you will be left where you were when you typed the ?.

HELP To read about any of the commands listed when you type ?, type help *command*) . For example, type MM> help type) for information on the type command.

For all commands that require a message number, you may also give several numbers separated by commas (for example, type 1,3,7) or, for consecutive messages, the first and last numbers separated by a colon (for example, delete 5:7).

CAUTION: Do not edit message files with Emacs. Always use MM commands to delete or edit messages.

7.2. MM - Lesson Two - Sending a Message

To use the send command, type the UNDERLINED words as follows:

MM> send↵
To: *Sting* ↵ To send a message to more than one person, separate each directoryname with a comma. You may capitalize directorynames.

Cc: *Mercury* ↵ This is for carbon copies. Type the directorynames of those you want to receive a carbon copy of the message (including yourself).

Subject: *Live Aid*↵
Message (End with ESCAPE or CTRL/Z.
Use CTRL/B to insert a file, CTRL/E to enter editor, CTRL/K to redisplay message, CTRL/L to clear screen and redisplay, CTRL/N to abort.):

I saw your act at the Live Aid concert...you were great! Branford Marsalis was especially wonderful on the saxaphone (as always). Let's get together soon.

To end your message, type ↑Z or ⇜.

Send Mode Commands

Once you type ↑Z or ⇜ to end your message, the prompt on the left side of the screen will be S> because you are in send mode. Now you may use any of the following commands (type ? to see additional commands that can be used here):

SENDING A MESSAGE

To send your message, type ↵. **You may edit your mm.init file so that you must type send↵ to send your message.**

ABORTING A MESSAGE

To abort your message, type quit↵.

SEEING A MESSAGE

To see your entire message before sending it, type display↵.

EDITING A MESSAGE

To edit your message in Emacs, type edit↵ and you will enter Emacs with the body of your message in the main buffer. You may edit your message and then return to MM by typing ↑X↑Z.

CHANGING THE FROM FIELD

Your name will automatically show in the From field. If you are sending a message for someone else, you may enter their name in the From field by typing from *Name*↵. The name does not have to be a directoryname. Your name will then appear in the message header in the Sender field. Do not type a colon when you are adding this field.

REPLYING TO A MESSAGE

To reply to a message you have received, type reply *#*↵. MM will prompt you, "Send reply for message # to:" and you may type ↵ to reply to the sender only, or type /a↵ to send a copy to everyone listed in the original message. (Type ? to see other options.) The **To**, **From**, and **Subject** fields will be completed by MM, and you will be prompted for **Text**. Type your reply and end the message with ↑Z or ⇜. Type send↵ to send it.

FORWARDING A MESSAGE

To send a copy of a message you have received to someone else, type forward *#*↵. You will be prompted for **To** and **Text**. If you want to include your own message along with the forwarded message, type your message here and end with ↑Z or ⇜. Type send↵ to send it.

REMAILING A MESSAGE

Remail allows you to send a copy of a message you have received to someone else without adding anything to the message. Type remail *#*↵. You will be prompted for To and your message will be sent immediately.

SAVING AND RESTORING DRAFT MESSAGES BEFORE SENDING

If you have created a message but you do not want to send it immediately, you may store it as a file in your directory and then retrieve and send it at a later time. To store your message before sending it, type (at the S> prompt)

S> save-draft *filename*↵
S> quit↵

To retrieve your stored message, type

MM> restore-draft *filename*↵

and you will be in the text field of your message. You may add to your message or type ↑Z or ⇜ to end it, and then edit or send it.

7.3. MM - Lesson Three - Managing Your Mail

DELETING MESSAGES

To delete a message, type delete *#*↵ . To delete a sequence of messages, type delete and the numbers of the first and last messages in the sequence separated by a colon (delete 10:15↵ deletes messages 10 through 15). To delete messages not in sequence, separate each message number with a comma. Deleted messages can still be retrieved with the undelete command as long as they have not been expunged.

UNDELETING MESSAGES

To undelete a deleted message, type undelete *#*↵. You may also undelete more than one message at a time.

EXPUNGING DELETED MESSAGES

To **permanently** remove all deleted messages, type expunge↵ . Your remaining messages will be renumbered and you will be left in MM.

PRINTING MESSAGES

To print a lineprinter (hardcopy) listing of a message, type list *#*↵ .

MOVING MESSAGES TO OTHER FILES

This command is useful for collecting related messages into individual files. To move messages from your mail.txt into another message file, type move *filename #*↵. If you have just read a message, you can move it without specifying the message number.

SEEING OTHER MESSAGE FILES

To read a mail file other than mail.txt into MM, type get *filename*↵ . To return to your mail.txt, type get↵ .

8. CCA EMACS

CCA Emacs[3] is a text editor. It is a **screen editor**. CCA Emacs commands start with either the Escape Key or the Control Key. When you use the Escape Key, type it once, release it, then type the next character. When you use the Control Key, hold it down while you type the other character. Anything you type without the Escape Key or the Control Key will be entered as text. The cursor shows where text will be entered if you type, and the cursor can be moved anywhere on the screen.

Near the bottom of the screen is a line called the **Mode Line**. It shows the name of the buffer and file you are currently working on, whether the file has been changed since you last updated and saved it (indicated by an asterisk after the filename), where you are in the file, and what mode you are in.

The area of the screen below the mode line often shows where you are in a command. For example, whenever you type the Escape Key, the characters **M-** appear. When you see that, you know that CCA Emacs is going to consider the next character you type to be a command. It is important to pay attention to this part of the screen.

If you neglect to type a carriage return at the end of a line of text, the line will wrap around and whatever doesn't fit on the first line will print on the next line. Emacs will still consider this one line, however, and the way it shows this is by printing a ! (exclamation point) at the end of the top line.

A complete CCA Emacs manual is available from Computer Corporation of America, Four Cambridge Center, Cambridge, MA 02142.

---

[3]

8.1. CCA Emacs - Lesson One

Key:
↑ = Control Key
⇦ = Escape/Alt Key
⤶ = Carriage Return

ENTER EMACS ed⤶ . This is for HOBGOBLIN users. Other VAX machines may require that you type cemacs⤶ .

CREATING A NEW FILE

To create a new file, enter Emacs and start typing.

NAMING A NEW FILE

↑X↑W *newfilename*⤶ . This commands also writes out (saves) the file.

UPDATING/SAVING A FILE

↑X↑S . This saves the file you are working on under the same name as the original, and replaces (overwrites) the original file. If you want to save a new version but also keep the original, you should use the NAME NEW FILE command (↑X↑W) and give the new version a slightly different name.

EDITING AN EXISTING FILE

↑X↑F *filename*⤶ .

CONTINUING EMACS

To return to the last file you edited in Emacs (without having to read it in again and find your place), type ed -s⤶ .

LEAVING EMACS ↑X↑Z . Be sure that you have saved the file (if you want it) before exiting.

KILLING LINES AND REGIONS

To kill one line, go to the beginning of the line and type ↑K. To kill a region (anything longer than a line), first mark the beginning of the region to be killed by moving the cursor to the first character of the region and typing ↑@ (the @ won't appear on the screen). Then move the cursor one character past the end of the region to be killed and type ↑W. To retrieve what has been killed, type ↑Y.

INSERTING BLANK LINES

Typing a carriage return will insert a blank line. Also, placing the cursor under the first character in a line and typing ↑O will insert a blank line above that line. Be sure that the cursor is at the beginning of the line or ↑O will cause a line break instead of a blank line.

HELP

To read the help options, type ↑_ ? and follow the instructions to see information about command characters, functions, etc.

ABORTING A COMMAND

Type ↑G to abort a command that has not yet been completed.

CUSTOMIZING EMACS

A file called **.emacs_vars** may be created (or may already exist) in your directory. This file is for the purpose of setting certain features of Emacs to your own preference. You may edit this file with Emacs. A complete list of features is in the CCA Emacs Manual.

AUTOSAVE FILES

Every 500 characters you type, Emacs will automatically save your file under the original filename followed by a ~ (*filename~*). Most of the time you can ignore autosave files. However, if you discover that a file that you edited does not contain your most recent additions or changes, see if there is an autosave file. If so, it may contain a version of your file with some or all of your recent changes.

8.2. CCA Emacs - Lesson Two

SEARCHING FOR TEXT (Incremental Search)

To search forward in your file, type ↑S, then type the text you're searching for. To find successive occurrences, repeat ↑S, and to end the search type ↑H. To search backward in your file, use ↑R.

EXCHANGING TEXT (Query Replace String)

This command finds each successive occurrence of a string of text and allows you to decide whether or not to exchange it for another string of text. Type ⇋r *existing text* (give current string to be changed) ⇋ *new text* (give desired string) ⇋ to start the exchange. As each occurrence of existing text is found, you may have it replaced with new text by typing one space bar, skip it by typing the ← (or Del Key), or confirm all exchanges by typing !.

JOINING FILES (Insert File)

To combine two files, read one file into a buffer with ↑X↑F *filename*), move the cursor to the point in the file where you want to insert the second file, and type

⇋x insert file *filename*)

The second file will be inserted immediately below the cursor. With this command, you may not use the ⇋ to complete the filename.

MULTIPLE BUFFERS

A buffer is a working space in Emacs. While you are editing a file, it is in a text buffer. Each buffer holds one file and has the same name as the file except for the buffer called **main**. Main is the empty buffer you are in when you first enter Emacs. Emacs allows you to have multiple buffers (with files in them) and to switch back and forth between them. Each time you type ↑X↑F *filename*⇋) you are reading a file into a buffer. To see the names of the buffers and the names of the files in them, type ↑X↑B. Type ↑L to clear the list of buffer names and redisplay your file. To switch from one buffer to another, type ↑XB *buffername*).

REPEATING COMMANDS

Most commands can be given so that they repeat a specified number of times. To repeat a command, type ↑U, then type the number of times you wish to repeat the command, then type the command. For example, to move forward 1 word in a file type ⇋F, and to move forward 10 words type ↑U 10 ⇋F . This command has a default of 4, so type ↑U ⇋F to move forward 4 words.

SPLIT SCREEN

To display two files simultaneously, read in one file with ↑X↑F *filename*⇋) , then create a second window with ↑X2. Now you will see the same file in both windows but you may read in a second file with ↑X↑F *filename*⇋) . To switch back and forth between windows, type ↑XO, and to return to one window, type ↑X1 . The split screen is also useful for looking at different parts of the same file simultaneously.

FORMATTING TEXT

Type ⇋Q to format the paragraph that the cursor is in. The default width is 70 characters per line. If you would like to specify a different width, you can set the fill column by typing ↑U 75 ↑XF, where 75 is the actual number you want the width to be set to. Look below the mode line and you'll see that it says **Fill Column = 75** (or whatever number you set it to). Once the fill column is set, type ⇋Q to format the paragraph. This command fills but does not justify text. To justify text, type ⇋1 ⇋Q .

AUTO FILL MODE

Type ⇋X auto fill⇋) . When you are inserting text in auto fill mode, Emacs will automatically enter carriage returns when you reach the end of a line. Use the same command to turn auto fill mode off. When auto fill is on, you will see the word **Fill** in your mode line. You may have auto fill set by default already in your .emacs_vars file.

8.3. CCA Emacs - Command Chart

DELETING TEXT

↑D	Delete next character
⇋D	Delete next word
←	Delete previous character (or Del Key)
⇋←	Delete previous word (or ⇋ Del Key)

MOVING THE CURSOR

↑F	Move forward one character
↑B	Move back one character
⇋F	Move forward one word
⇋B	Move back one word
↑N	Move to next line
↑P	Move to previous line
↑A	Move to beginning of line
↑E	Move to end of line
⇋A	Move to beginning of sentence
⇋E	Move to end of sentence
⇋[	Move to beginning of paragraph
⇋]	Move to end of paragraph
⇋<	Move to beginning of file
⇋>	Move to end of file
↑V	Display next screen of text
⇋V	Display previous screen of text

MISC. COMMANDS

↑L	Clear screen and reprint it
↑T	Switch the two characters to the left of the cursor
⇋T	Switch two words. Put cursor under 1st character of 2nd word.
⇋C	Change first letter of word to uppercase (**NOTE:** This is not ↑C)
⇋U	Change word to uppercase
⇋L	Change word to lowercase
⇋S	Center line

9. SCRIBE - BASIC COMMANDS

Scribe[4] is a text formatting program. It is used for preparing text to be a report, manual, article, letter, or any other type of document. Scribe processes text files that have formatting directions (commands) in them, and produces new files that can be printed on a variety of output devices. Some examples of the types of formatting commands are the following: italicize (make the following text appear in italics), center (center the following text), and underline (underline the following text). Scribe can distinguish commands from text because every Scribe command starts with an at sign (@). Another Scribe command you should give indicates the output device (Penguin, Lineprinter, Diablo, Imagen, or terminal screen) on which the file created by Scribe is to be printed. Once you use a text editor to create a file that contains text and Scribe commands, Scribe can process it and produce a second file, formatted according to the commands in the original file that you created.

First, use a text editor to create a file that has text along with Scribe commands. The Scribe commands are typed as if they were ordinary text. Then leave the text editor, return to the Shell, enter Scribe, and give it the name of the file that you created. Scribe will read that file, interpret the commands, and create a second file by following the directions (commands) that you put in the first file. Now you may send the file that Scribe created to the appropriate output device.

Scribe is a complex program with many features that you may want to use. This chapter provides a general overview of Scribe. A complete Scribe manual is available from UNILOGIC, Ltd., 160 N. Craig St., Pittsburgh, PA 15213.

---

[4] "Scribe" is a registered trademark of UNILOGIC. Ltd., Pittsburgh, PA.

9.1. Filename Types

The filename type is the part of the filename that follows the . (period) in the filename. With Scribe files, the type is significant. **The file you create with Scribe commands in it should always have the type .mss** (for example, *testfile*.mss). Scribe processes the .mss file you create and produces a second file with the same name but a different type. The type depends on the printing device you specified in the .mss file. The list below shows the file type that Scribe associates with each printing device.

If Scribe detects any illogical commands while it is processing your .mss file, it will also produce an .err file, which contains a list of errors in the .mss file. When this happens, you must fix the problems in the .mss file yourself because **Scribe never alters the .mss file.** After you have fixed the .mss file, run it through Scribe again.

.mss — Source file. When you create a file with Scribe commands in it, give it this type. **This is the only file that you name or edit.**

.doc — Terminal file. This is the default. If you don't give a device command (or if you use @device[file]), Scribe will produce a file with the type .doc. You can print this file on your terminal screen using the cat command at the Shell.

.prs — Penguin file. When you use @device[penguin], Scribe will produce a file with the type .prs. The .prs file must be printed on the Penguin.

.lpt — Lineprinter file. When you use @device[lpt], Scribe will produce a file with the type .lpt. The .lpt file must be printed on the Lineprinter.

.pod — Diablo file. When you use @device[diablo], Scribe will produce a file with the type .pod. The .pod file must be printed on the Diablo with the Podtyp program.

.imp — Imagen file. When you use @device[imprint10], Scribe will produce a file with the type .imp. The .imp file must be printed on the Imagen printer.

.err — Error file. If Scribe finds any errors while it is processing your .mss file, it prints them on your screen while it works and also produces an .err file that you can read with a text editor. Pay close attention to these error messages.

.otl — Outline file. This file contains an outline of the corresponding .mss file's chapters, sections, and subsections. Scribe creates an .otl file only for certain kinds of documents (for example, articles and manuals).

9.2. Global Commands

Global commands affect the entire file. If these commands are used, they must appear at the beginning of your .mss file. The three most often used Global commands are **make**, **device**, and **style**.

The @Make Command

If you use the @make command, it should be the first line of your .mss file. This command indicates the kind of document you are creating and should be followed by one of the options shown enclosed in delimiters:

@make[text] This is the default. If you don't give a make command, this is what Scribe will use. This is used for most documents that are not letters or slides, and do not require numbered chapters, or a table of contents.

@make[article] An Article has three levels of numbered headings: Section, Subsection, and Paragraph, as well as Appendix and AppendixSection. It also has a table of contents.

@make[report] A Report has a title page, numbered chapters, sections, subsections, and a table of contents.

@make[manual] Manual is like report, but can also have an index.

@make[slides] Slides uses a large font and appropriate line spacing for overhead transparencies.

@make[letterhead]
Letterhead produces a business letter format. (Use @make[isiletterhead] for ISI stationery.) This will not produce letterhead paper. You must print the letter on the Penguin or the Imagen, then copy it onto letterhead stationery. Or, print the letter on the Diablo, directly onto letterhead stationery.

The @Device Command

The second line of a Scribe .mss file should be the @device command. If there is no @make command, the @device command should be the first line of the file. This command tells Scribe to create a file that can be printed on a specified device. It should be followed by one of the options shown, enclosed in delimiters:

@device[file] This is the default. If you don't give a device command, Scribe will create a .doc file. You may print a .doc file on your terminal screen with either the **cat** or the **more** command at the Shell.

@device[lpt] This tells Scribe to create an .lpt file. The .lpt file should be printed on the Lineprinter with the print command at the Shell.

@device[diablo] This tells Scribe to create a .pod file. This file should be printed on the Diablo with the Podtyp program.

@device[penguin]
This tells Scribe to create a .prs file. This file should be printed on the Penguin with the xp command at the Shell.

@device[imprint10]
This tells Scribe to create an .imp file. This file should be printed on the Imagen with the lpr command at the Shell. **Note:** You must type setenv PRINTER isi-imagen at the Shell before using the lpr command.

The @Style Command

Style commands allow you to change the appearance of the entire document. If they are used, Style commands should be grouped together immediately following the @device command. Some useful Style commands are:

@style[spacing 2] Makes double-spaced text.

@style[justification off] Makes filled text with an unjustified right margin.

@style[indent 0] Causes the first line of each paragraph to be flush left.

@style[indent 5] Causes the first line of each paragraph to be indented five spaces.

9.3. Useful Scribe Commands

Most commands can be in either of two forms:

1. The command followed by **delimiters** surrounding the text to be affected, for example, @center[Scribe Commands], or

2. The command delimited by **begin** and **end**, for example,
   @begin[center]
   Scribe Commands
   @end[center]

The result of the two examples is the same:

Scribe Commands

Delimiters must match each other within a command. The delimiters you may use are: (...), [...], {...}, <...>, '...', and "..." . Choose your delimiters carefully so that the text or figures inside the delimiters do not include the particular delimiters used.

A blank line	A blank line indicates the end of a paragraph.
@center	This command centers text.
@i	This command italicizes text.
@u	This command underlines text. (See page 12 of the Scribe manual for underlining options.)
@format	Format produces text that is not filled or justified but appears exactly as you typed it. The text within format stays in the same font as the body of the text. Other Scribe commands within format will still be recognized.
@verbatim	This command is similar to format, but the text in verbatim is printed in a fixed-width font (similar to a typewriter).
@quotation	This command produces text that is single-spaced, filled and justified, and with wider right and left margins (the margin is the white area on each side of the text).
@verse	Verse produces results similar to quotation but does not fill and justify.

@itemize Itemize widens both the right and left margins, fills and justifies the text, and places a tick mark (–) at the beginning of the first line of each new paragraph. Separate paragraphs as you type them with a blank line.

@enumerate Enumerate is almost the same as itemize except that numbers instead of tick marks are placed before each paragraph.

@description Description was used to format the text on this page. The word or phrase to be described is typed, followed by @\ (Scribe's tab indicator), followed by the text of the description. Each word or phrase and its description are considered a paragraph, and paragraphs are separated with a blank line.

@blankspace[n lines] or [n inches]
This command tells Scribe to insert blank lines in your text. Use @blankspace[5 lines] to insert five blank lines. You may also use @blankspace[3.4 inches] to insert a measured vertical space.

@ Use this to indicate a tab (instead of the tab key on your terminal). Be sure to use the \ (back slash) and not the / (forward slash). It is often necessary to use the format or verbatim command around the text with tabs so that the text is not filled and justified. (See page 48 for an example of using tabs.)

@> This command pushes whatever text follows it to the right margin (flush right). When @> is used with a tab, the text will be flushed against the tab instead of the right margin. It is often necessary to use the format or verbatim command around the text with this command so that the text is not filled and justified. (See page 48 for an example of using this with tabs.)

@ space bar Typing one @ sign followed by one space creates one and only one space in the text. Typing @ @ @ (three @ signs, each followed by a space) creates exactly three spaces in your text.

@@ Typing two @ signs together in the .mss file causes one @ sign to be printed in your text.

@newpage This command causes a page break.

@* This command causes a line break.

@group Group prevents the text from being broken between pages. If you have a paragraph or chart that is being broken up between pages in an awkward way, this command will cause it to be printed on one page. Usually this should be used in the @begin[group] ... @end[group] command format.

Remember to start all Scribe commands with an @ sign.

When using commands with @begin and @end, it is good to put the @begin[command] and @end[command] on lines by themselves. Every @begin must have an @end or Scribe will generate an error message.

Headings for documents with and without a table of contents are listed in the section on "Titles, Sections, and the Table of Contents" of the Scribe manual.

9.4. Sending an .Mss File through Scribe

When your .mss file is ready to be sent through Scribe, type the following from the Shell (underlined words are what you type):

```
% scribe)
Scribe 3C(1312)-3  Copyright ©  1981  UNILOGIC, Ltd.
Welcome to UNIX Scribe.
        Help in /usr/scribe/help.  Report bugs to ACTION@USC-ISI.
* filename.mss)
[Processing filename.mss
    [Device "penguin"]
    [Document type "text"
        [FontFamily helvetica10]
    ]

[1  2  3]
4

**filename.prs for device PENGUIN has 4 pages.

%
```

In this example, Scribe processed the .mss file, prepared a .prs file for the Penguin, and returned to the Shell. The .prs file can now be sent directly to the Penguin with the xp command.

Modifying the Device Command

You may indicate the device at the time you enter Scribe and give the filename. This is useful if you have not given the @device command in your .mss file, or if you would like to indicate a device other than the one given in your .mss file. After you enter Scribe, type:

filename -penguin	Scribe will create a .prs file for the penguin
filename -imprint10	Scribe will create an .imp file for the imagen
filename -diablo	Scribe will create a .pod file for the diablo
filename -lpt	Scribe will create an .lpt file for the lineprinter
filename -file	Scribe will create a .doc file for the terminal

9.5. Example of .Mss and .Prs File

Sample .mss file (see next page for the .prs version). .mss file

```
@device[penguin]
@style[indent 0]
@majorheading[Fun with Scribe]
The purpose of this file is to demonstrate the results of using
a few of Scribe's simple commands.
@i[Here is an example of the use of the italics command.]
@u[Here is an example of the use of the underline command], and
here is an example of the
use of the @center[center command.]
Look at this page to see the command or @b[.mss] file, and
look at the next page for the @b[.prs] file (the file Scribe
produced by following the commands in the .mss file).

The @b[itemize] and @b[enumerate] commands are also very simple
to use.
Use them with the @@begin and @@end commands, and separate each
paragraph with a blank line.  The following was produced with
the @@enumerate command:
@begin[enumerate]
The only difference between @b[itemize] and @b[enumerate] is that
@b[enumerate] numbers each paragraph and @b[itemize] puts a tick
mark by each paragraph.

It is very important to remember to @b[end] any command that was
started with a @b[begin].  If you do forget, there will be at
least one error message when the file is run through Scribe.
@end[enumerate]
@blankspace[1 line]
@begin[description]
Example@\The description command separates a word or phrase
from text.  If the word or phrase overlaps the tab, the text
will start on the next line.  Separate paragraphs with a blank
line.
@end[description]
@begin[quotation]
The Quotation command causes both margins to be indented,
fills and justifies the text, and
inserts a blank space before and after the text delimited by
the command.  The Verse command is similar, but does not
fill and justify.
@end[quotation]
```

.prs file

Fun with Scribe

The purpose of this file is to demonstrate the results of using a few of Scribe's simple commands. *Here is an example of the use of the italics command.* Here is an example of the use of the underline command, and here is an example of the use of the center command.

Look at this page to see the command or **.mss** file, and look at the next page for the **.prs** file (the file Scribe produced by following the commands in the .mss file).

The **itemize** and **enumerate** commands are also very simple to use. Use them with the @begin and @end commands, and separate each paragraph with a blank line. The following was produced with the @enumerate command:

1. The only difference between **itemize** and **enumerate** is that **enumerate** numbers each paragraph and **itemize** puts a tick mark by each paragraph.

2. It is very important to remember to **end** any command that was started with a **begin**. If you do forget, there will be at least one error message when the file is run through Scribe.

Example — The description command separates a word or phrase from text. If the word or phrase overlaps the tab, the text will start on the next line. Separate paragraphs with a blank line.

The Quotation command causes both margins to be indented, fills and justifies the text, and inserts a blank space before and after the text delimited by the command. The Verse command is similar, but does not fill and justify.

9.6. Tabs

.mss file

Sample .mss file (see next page for the .prs version).

```
@subheading[Setting Tabs]

@begin[verbatim]
@tabs[1 inch, 2 inches, 3 inches, 4 inches]
@\344@\4859@\5938@\495
@\3988@\3975@\384@\394
@\@\@\397@\2987
@tabclear

@tabdivide[5]
@\234@\8796@\9876@\986
@\2368@\123@\6797@\23
@\37545@\2346@\3759@\234
@tabclear

@subheading[Tabs Used with the Flushright Command]

@tabs[1.5 inches, 3 inches, 4.5 inches]
@>$10100.00@\@>$2000.00@\@>$@ @ 3.00@\
@>10.00@\@>150.00@\@>130.00@\
@>320.00@\@>4500.00@\@>244.00@\

@tabclear
@tabdivide[4]
@>$10100.00@\@>$2000.00@\@>$@ @ 3.00@\
@>10.00@\@>150.00@\@>130.00@\
@>320.00@\@>4500.00@\@>244.00@\
@end[verbatim]


@subheading[Setting Tabs Visually]

@begin[format]
@tabclear
These people will arrive at 10:00 a.m.:   @↑Lisa Trentham
@\Sheila Coyazo
@\Lori Holzer
@end[format]
```

Setting Tabs

344	4859	5938	495
3988	3975	384	394
		397	2987

234	8796	9876	986
2368	123	6797	23
37545	2346	3759	234

Tabs used with the Flushright Command

$10100.00	$2000.00	$ 3.00
10.00	150.00	130.00
320.00	4500.00	244.00

$10100.00	$2000.00	$ 3.00
10.00	150.00	130.00
320.00	4500.00	244.00

Setting Tabs Visually

These people will arrive at 10:00 a.m.: Lisa Trentham
Sheila Coyazo
Lori Holzer

9.7. Scribe Practice

Homework #1

Follow these steps to Scribe

1. Use a text editor to create a file that contains some Scribe commands (try @device, @center and @enumerate).

2. Exit from the editor naming the file *filename*.mss. Always give a file containing Scribe commands the type .mss.

3. You are now at the Shell and ready to send the file you have created through Scribe. Type scribe↵ and then type the name of the file you created (*filename*.mss). Scribe processes that file by interpreting the commands and creates a new file with the same name but a different type. The type Scribe gives the file depends on the device command you used in your .mss file (lineprinter, penguin, imprint10, file, or diablo).

4. When finished, Scribe leaves you at the Shell. Now you can give the Shell command to send the file that Scribe produced to the appropriate output device. For the Lineprinter, type print *filename*.lpt↵. For the Penguin, type xp *filename*.prs↵ . For viewing on your terminal, type cat *filename*.doc↵ .

5. Examine the output to see if you got the results you expected. If you didn't, check your .mss file to determine what went wrong. Correct your error, and then run the .mss file through Scribe again.

Homework #2
Create this file

Practice Makes Perfect??

Use a text editor to create a file that has text along with Scribe commands. The Scribe commands are typed as if they were ordinary text. Then leave the text editor, return to the Shell, enter Scribe, and give it the name of the file that you created. Scribe will read that file, interpret the commands, and create a second file by following the directions (commands) that you put in the first file. Now you may send the file that Scribe created to the appropriate output device. In this exercise, several commands will be used:

1. @blankspace
2. @enumerate
3. @description
4. @u
5. @i
6. @center

It will be your HOMEWORK

TO USE THEM CORRECTLY!

Example The Description command creates an environment. This environment produces paragraphs with heading words in the margin. This environment is designed for reference tables containing a list of words or terms, each accompanied by a line or paragraph of description.

Second header The beginning of a new paragraph marks the beginning of a new description item.

Guess which commands are used here!!

THE END

Homework #3
Create this file

Scribe the Easy Way

by I. Format Freely

July 8, 1980

Scribe is used to create formatted text. The file you create with a text editor contains text along with Scribe commands indicating how you would like the text to be formatted. Each Scribe command starts with the @ sign.

Most commands can be in either of two forms:

1. The command followed by **delimiters** surrounding the text to be affected, for example.
   @center[Scribe Commands], or

2. The command delimited by **begin** and **end**, for example,
   @begin[center]
   Scribe Commands
   @end[center]

The result of the two examples is the same:

Scribe Commands

You can put a command into your manuscript file that tells Scribe which printing device your file is intended for. The following chart shows the command to use for each device.

@device[file]	creates .doc	file for the terminal screen
@device[penguin]	creates .prs	file for the Penguin
@device[imprint10]	creates .imp	file for the Imagen
@device[diablo]	creates .pod	file for the Diablo
@device[lpt]	creates .lpt	file for the Lineprinter

It is not necessary to give a device command if your file is intended for the terminal screen. If no device command is given, the default is to create a file for the terminal screen.

9.8. Using Scribe for Letters

To create letters formatted for letterhead stationery, use @make[letterhead]. (Use @make[isiletterhead] for ISI letterhead.)

Device options are:

- @device[diablo]
- @device[penguin]
- @device[imprint10]
- @device[lpt] Use this for proofreading a letter.
- @device[file] Use this for proofreading a letter.

The command @string[name=" "] (shown on page 54, in the sample .mss file) is useful only for letters with more than one page. After page one, the name given in the string (between the quotes) will appear at the top left of each succeeding page along with the page number in the center and the date on the right.

For paragraph indentation, use @style[indent 5] to indent five spaces, or @style[indent 0] for no indentation.

Penguin Letter If you have created a .prs file for the Penguin, you will use the xp command from the Shell to send the file to the Penguin. After the file has been printed on the Penguin, you must photocopy it onto letterhead stationery.

Diablo Letter The default font for the Diablo is **Elite**. If you plan to use the Elite print wheel, it is not necessary to have a font command in your .mss file. Be sure the pitch switch on the Diablo is set to 12. If you are going to use the Pica print wheel, use the command @style[fontfamily = pica] in your .mss file. For this print wheel the pitch switch should be set to 10. After Scribe produces the .pod file, use the Podtyp program on the Diablo to print your letter directly onto letterhead stationery.

Sample .Mss File for Penguin letter (see next page for the .prs version).

```
@make[isiletterhead]
@device[penguin]
@string[name="Mr. Stevie Wonder"]
@begin[address]
Mr. Stevie Wonder
111 Super Street
Numberone, Everywhere  11111
@end[address]
@begin[body]
Dear Stevie,

Your last album, "Hotter than July," was truly hot and I loved
every minute of it.  "Rocket Love" was really spacy and all
the tracks were special in some way.  The problem is, I played
it sooo much, I'm sick of it and now I can't wait for your next
one.  Based on past experience, I'm afraid that you will keep
me in agony for another year and I just want you to know that
I will come unglued if you do!

I also want to know when you're going to give another concert
in Los Angeles.  All the kids here are saving their money for
that magic day when tickets for your concert go on sale and
they can wait in line all night to be the first to get them.

Don't forget, if you ever need an enthusiastic president for
your Stevie Wonder Fan Club, you know you can call on me.

@end[body]
Yours forever,



Ms. Fan Ardent
@begin[notations]
FA/fa


cc:  @↑Mr. Peter Allen
@\Ms. Aretha Franklin

Enc.  "A Groupie's View of Stevie Wonder"
@end[notations]
```

October 31, 1981

Mr. Stevie Wonder
111 Super Street
Numberone, Everywhere 11111

Dear Stevie,

Your last album, "Hotter than July," was truly hot and I loved every minute of it. "Rocket Love" was really spacy and all the tracks were special in some way. The problem is, I played it sooo much, I'm sick of it and now I can't wait for your next one. Based on past experience, I'm afraid that you will keep me in agony for another year and I just want you to know that I will come unglued if you do!

I also want to know when you're going to give another concert in Los Angeles. All the kids here are saving their money for that magic day when tickets for your concert go on sale and they can wait in line all night to be the first to get them.

Don't forget, if you ever need an enthusiastic president for your Stevie Wonder Fan Club, you know you can call on me.

Yours forever,

Ms. Fan Ardent

FA/fa

cc: Mr. Peter Allen
Ms. Aretha Franklin

Enc. "A Groupie's View of Stevie Wonder"

10. DIABLO PRINTER

The Diablo printer is similar to a terminal but uses paper instead of having a screen. You must first login (through the Micom) on the computer where the file you are going to print exists.

If you are starting a long job on the Diablo, check the ribbon first so you don't run out while your job is printing.

The **Local** Key (found to the left of the keyboard) causes the Diablo to function like a typewriter, not a terminal. If you need to type something in addition to your file once you are at the Diablo (like an underline or address), press the local key down. Also, if the Diablo is not responding to commands, check this key to make sure that it is not pushed down.

When you are finished using the Diablo, be sure to logout and turn the power switch off. If the Diablo is left on too long, it may overheat.

10.1. Podtyp – Printing Scribe Files on the Diablo

Key:
) = Carriage Return

A Scribe-created .pod file is printed on the Diablo using the Podtyp program. To use Podtyp, put a sheet of scratch paper in the Diablo, login, and type:

%1 podtyp)
Type file name or ? for help
filename.pod)

Replace your scratch paper with a clean sheet. Type) and Podtyp will roll the paper up 7 lines (as if you had typed 7 carriage returns), then it will start printing your file. The Diablo will pause at each new page so you may insert a piece of paper, then type) to start printing. When the last page has been printed, insert your scratch paper and type q) to quit from Podtyp, or one of the following commands:

a	reprint the page just printed
b	set current page to 0 (start over)
e	set current page to last page
+n	set page to current page +n (n is a number)
–n	set page to current page –n
=n	set current page equal to *n*
g	turn continuous mode on
n	ready to print another file
s	turn continuous mode off
q	quit from podtyp program
?	help (to get this message)

10.2. Dcopy - Printing Text Files on the Diablo

Key:
↑ = Control Key
) = Carriage Return

To print a text file on the Diablo using the Dcopy program, type:

dcopy *filename*)

insert a fresh sheet of paper and hit RETURN to begin

Replace your scratch paper with a clean sheet and type) to begin printing.

Approximately 55 lines will fit comfortably on each page. To print a file that is longer than one page, insert ↑L on a line by itself (in Emacs), wherever you want a page break. To insert ↑L, type ↑Q↑L and just the ↑L will appear. The ↑L is a pagebreak command and will not be printed when you run Dcopy.

When your file has finished printing, insert your scratch paper and type) to quit from Dcopy.

11. FTP - TRANSFERRING FILES

The following example shows a user who is logged into HOBGOBLIN, **sending** a file to ISIB, and then **getting** a file from ISIB.

5% **ftp isib**↵
Connected to usc-isib.arpa.
220 usc-isib Server FTP
Name (usc-isib.arpa:moses): ↵ **(Type a directoryname here only if it's different than the logged in directoryname.)**
Password (usc-isib.arpa:moses): ***password***↵
331 Enter PASS Command
230 User Logged in
ftp> **?**
Commands may be abbreviated. Commands are:

A list of commands will be printed in response to the ? command.

ftp> **send *filename***↵
200 Host 10.1.0.52, port 2642
125 Storing "filename" started okay
226 File transfer completed okay
447 bytes sent in 0.14 seconds (3.1 Kbytes/s)

ftp> **get *filename***↵
200 Host 10.1.0.52, port 2644
150 Retrieval of "filename" started okay
226 File transfer completed okay
968 bytes received in 0.17 seconds (5.6 Kbytes/s)
ftp> **bye**↵
221 Toodles, call again
7%

12. TELNET – ACCESSING OTHER COMPUTERS

The following example shows a user who is logged into HOBGOBLIN, using Telnet to login to ISIB, then returning to HOBGOBLIN.

1% **telnet isib**↵
Trying...
Connected to usc-isib.arpa.
Escape character is '↑]'.

ISI-System-B, TOPS-20 Monitor 5(7270)
There are 62 + 10 jobs and the Load Average is 3.22

@***directoryname password***↵
@
@**logo**↵
Killed Job 83, TTY 214,
at 23-Oct-84 14:37:24, Used 0:00:00 in 0:00:12
Connection closed.
2%

If you logout of a Telnet job and you are not immediately returned to your original logged in job, type ↑] to return to Telnet (you must type it twice if you are using KERMIT on an IBM PC) and then type **quit**↵.

To see all of the Telnet commands, type:

1% **telnet**↵
telnet>**?**

To leave Telnet, type **quit**↵

Appendix I
HOBGOBLIN

I.1. Hostname and Net Address

REACHING HOBGOBLIN THROUGH THE MICOM

Before you login, the Micom port selector will ask you to "ENTER CLASS." To open a connection to Hobgoblin, you must respond by typing h↵. As soon as you see GO you must type ↵↵↵ until you see **Login:** printed. Then you may login. If you don't type the ↵↵↵ when you see GO, you will lose your connection.

REACHING HOBGOBLIN THROUGH TELNET

To reach Hobgoblin with the Telnet program, type telnet hobgoblin↵ (see the chapter on Telnet in this manual for further instructions).

NET ADDRESS OF HOBGOBLIN

The complete net address of Hobgoblin is **isi-hobgoblin.arpa**. To send mail to someone on Hobgoblin from another ISI machine, type *directoryname*@hobgoblin in the **To** field of the message. If you are logged in on Hobgoblin and you are sending mail to someone else on Hobgoblin, you only need to give the directoryname.

I.2. Aliases and the .Login and .Cshrc Files

Your .login file and .cshrc file are read by the System when you login. These files have been written to customize some UNIX commands and functions. This section describes the aliases that may have already been set in your directory on Hobgoblin.

kk,logo kk↵ or logo↵ are the same as the logout command.

ty, type ty *filename*↵ will show the contents of a file. It is the same as cat↵.

del, und, exp
del *filename*↵ will put the file in a temporary directory rather than removing it permanently. You may und *filename*↵ to retrieve it, or exp↵ to permanently remove all deleted files. You may see all of your deleted files (before they are expunged) by typing dir del↵. **NOTE:** This only applies to files removed with the del or clean commands.

clean clean↵ deletes your Emacs autosave files (see chapter on Emacs). It is like the del *filename*↵ command. Files may be retrieved with und *filename*↵.

back back↵ connects you to the last directory you were in.

dir dir↵ gives a list of files in your directory indicating subdirs with a / and programs with a * (as with the ls -F command). It also shows the protection status of your files.

push push *directoryname*↵ allows you to connect to another directory or subdirectory and is the same as typing cd *directoryname*. The advantage to the push command is that it keeps track of the last directory you were in and when you type **pop**↵ you are returned to that directory.

rd rd↵ shows your list of files sorted by read date.

wd wd↵ shows your list of files sorted by write date.

tn tn *hostname*↵ is the same as telnet *hostname*↵.

cont cont↵ is the same as fg to continue a stopped job.

hist hist↵ works the same as history to show the last twenty shell commands.

da da↵ works the same as date to show the day and time.

Appendix II
SAMPLE .CSHRC FILE

If your directory contains files called **.cshrc** and **.login**, the System will read those files when you login and incorporate aliases and other instructions in them. The following files are copies of the .cshrc and .login files I use, and are intended as examples. Terminal type is defined as **h19** for over 4800 baud (good for IBM PC's), and **hp** for 4800 baud or less. The editor is defined as CCA Emacs. Lines beginning with **#** are comment lines. **NOTE:** To fully understand these files, consult more advanced UNIX documentation.

```
set history = 25
set prompt = "'hostname' \!% " notify junk = $home/.dlsave
umask 007
alias chmod 'chmod \!* ; ls -ld \!:2*'
alias push pushd
alias pop popd
# aliases for new del/undel system
alias clean 'del *.BAK *.CKP *~ *.bak .[A-z]*~'
alias und 'undel \!*'
alias exp '/usr/local/expunge \!*'
alias dir \
 'if ( "\!:*" = = del ) dls -laF;' \
 'if ( "\!:*" != del ) ls -lF \!:* ;'
alias bb bboard
alias mail inc
alias ap apropos \!\*
alias kk logout
alias logo logout
alias wd ls -Ftl \!\*
alias tn telnet
alias type cat
alias cont fg
alias hist history
alias da date
alias ed cemacs
alias emacs /usr/bin/cemacs
alias gemacs /usr/bin/emacs
```

Appendix III
SAMPLE .LOGIN FILE

```
set path = (. \
~/bin      \
/usr/new   \
/usr/ucb   \
/bin       \
/usr/bin   \
/usr/local \
/usr/new/mh \
/usr/stanford/bin \
/usr/games \
)
set mail = (60 /usr/spool/mail/$user)
set cdpath = ($user)
setenv CZ "-n "'"'"$user at 'hostname'"'"'
setenv VERSIONS 4
set ignoreeof time = 15
alias setchr "stty -nohang intr ^c kill ^u erase \ crt crterase"
alias ts \
 'stty -tilde ;'\
 'tset -Q -s \!* > /tmp/$$ ;' \
 'source /tmp/$$ ; rm /tmp/$$ ;' \
 'if ( $TERM = = h1520 ) stty tilde ; ' \
 'if ( $TERM = = 2640 ) stty nl1 ;' \
 setchr
 # initial termcap setup
if ( $?TERMCAP = = 0 ) then
  tset -Q -s \
    -m 'dialup< = 1200:?hp' \
    -m 'dialup = 4800:hp' \
    -m 'dialup>4800:h19' \
    -m 'decwriter:dw2' \
    hp \
     > /tmp/$$
 source /tmp/$$
 rm /tmp/$$
 if ( $TERM = = 2640 ) stty nl1
endif
setchr
```

Index

END

FILMED

11-85

DTIC